



UniFly

Jour Journey, Our Priority!

Relazione Progetto Basi Di Dati

Autore: **Giulio Pedicone**

Matricola: **1000031068**

A.A. 2022-2023

INDICE

<u>Descrizione del progetto</u>	<u>[1]</u>
<u>Specifiche sui dati</u>	<u>[1]</u>
<u>Glossario dei termini</u>	<u>[3]</u>
<u>Definizione delle operazioni</u>	<u>[4]</u>
<u>Progettazione concettuale</u>	<u>[5]</u>
<u>Schema scheletro</u>	<u>[5]</u>
<u>Raffinamenti</u>	<u>[6]</u>
<u>Schema finale</u>	<u>[8]</u>
<u>Vincoli non esprimibili dal diagramma E-R</u>	<u>[9]</u>
<u>Dizionario dei dati</u>	<u>[10]</u>
<u>Dizionario delle relazioni</u>	<u>[11]</u>
<u>Tabella dei Volumi</u>	<u>[12]</u>
<u>Tabella delle Frequenze</u>	<u>[13]</u>
<u>Schemi delle Operazioni</u>	<u>[14]</u>
<u>Traduzione verso il modello Relazionale</u>	<u>[19]</u>
<u>Schema Logico</u>	<u>[20]</u>
<u>Schema Fisico</u>	<u>[21]</u>
<u>Trigger in SQL</u>	<u>[22]</u>
<u>Vista in SQL</u>	<u>[23]</u>
<u>Query in SQL</u>	<u>[24]</u>

Note:

Il presente documento è concepito esclusivamente a fini didattici, pertanto, il progetto in esso delineato non mira in alcun modo ad essere realizzato nell'ambito della pratica effettiva.

É possibile consultare la seguente repository **GitHub** per una migliore visualizzazione degli schemi sottostanti:

<https://github.com/giuliopedicone02/Database>

Descrizione del Progetto

Il database "**UniFly**" è un sistema progettato per gestire e coordinare le attività di una **rete aeroportuale** e le operazioni delle compagnie aeree che operano al suo interno.

Il database tiene traccia di informazioni essenziali relative agli **aeroporti**, alle **compagnie aeree**, ai **voli**, ai **passengeri**, agli **aerei**, al **personale di bordo** e ai **servizi** offerti durante i voli.

Grazie a questo sistema, è possibile offrire un'esperienza di viaggio ottimizzata ai passeggeri, garantire una corretta pianificazione dei voli, gestire l'assegnazione dei **posti** e dell'equipaggio e monitorare i servizi offerti durante i viaggi.

Specifiche sui dati

Dati di **carattere generale**:

Si vuole progettare un DBMS atto a gestire una **rete aeroportuale**. Si vogliono identificare diverse entità quali **passengeri**, **voli**, **aerei**, **personale di bordo** e **servizi offerti**.

Dati sugli **Aeroporti**:

Per gli **aeroporti**, rappresentiamo alcuni dati come il nome, la posizione geografica, il codice ICAO (International Civil Aviation Organization) e un ID univoco per ciascun aeroporto.

Questi dati sono essenziali per organizzare e gestire i voli in partenza e arrivo da ogni aeroporto della rete.

Dati sulle **Compagnie Aeree**:

Per le **compagnie aeree**, rappresentiamo alcuni dati come il nome della compagnia e il suo codice ICAO. Questi dati consentono di tenere traccia delle diverse compagnie coinvolte nelle operazioni e nelle prenotazioni.

Dati sui **Passeggeri**:

I **passeggeri**, vengono identificati da un ID univoco e rappresentano coloro che prenotano un volo. Vengono raccolti dati quali il nominativo completo ed il numero di telefono.

Dati sui **Voli**:

Ogni **volo** ha un ID univoco e include il numero del volo, la data e l'ora di partenza, i riferimenti agli aeroporti di partenza e arrivo e l'aereo che opera il volo. Questa tabella è fondamentale per la pianificazione dei voli ed il monitoraggio delle operazioni aeroportuali.

Dati sugli **Aerei**:

Per gli **aerei**, rappresentiamo alcuni dati come il produttore, il modello, la capacità di passeggeri, l'anno di fabbricazione e il riferimento alla compagnia aerea proprietaria.

Dati sui **Posti**:

Ogni **posto** ha un ID univoco e include il numero del posto e la classe (Economy, Business, First). Ogni posto è collegato a un aereo specifico, consentendo una gestione accurata della disposizione dei posti sugli aerei.

Dati sull' **Equipaggio**:

Ogni membro dell'**equipaggio** ha un ID univoco e include il nome, il cognome, la data di nascita e il ruolo (pilota, copilota, assistente di volo)

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Aeroporto	Località di decollo ed atterraggio dei voli	Terminal	Volo
Compagnia Aerea	Ditta che offre i servizi di trasporto aereo	Linea Aerea	Aereo
Passeggero	Viaggia a bordo di un aereo	Cliente	Prenotazione
Volo	Operazione di trasporto aereo	Viaggio	Aeroporto, Aereo, Prenotazione, Equipaggio, Servizio
Aereo	Mezzo di trasporto	Velivolo	Compagnia Aerea, Volo, Posto
Posto	Area di un aereo designata alla seduta dei passeggeri	Poltrona	Prenotazione, Aereo
Prenotazione	Acquisto di un viaggio	Booking	Passeggero, Volo, Posto
Equipaggio	Gruppo di persone che opera e gestisce l'aereo	Staff	Volo
Servizio	Servizi destinati al comfort dei passeggeri	Extra	Volo

Definizione delle operazioni

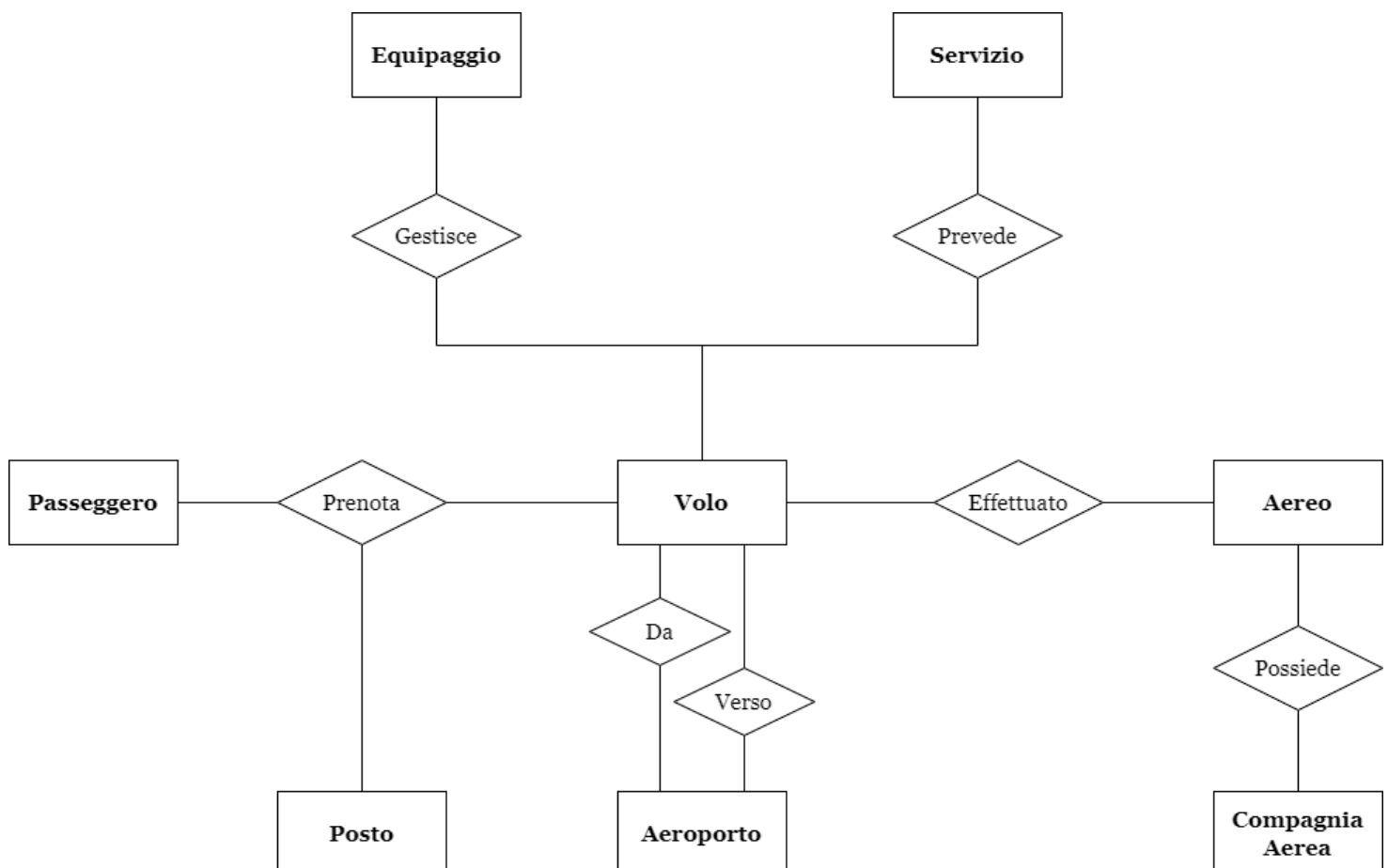
Indice	Operazione
1	Inserire una nuova prenotazione
2	Visualizzare i voli prenotati da un passeggero
3	Vedere il numero di aerei per ogni compagnia aerea
4	Trovare il volo che prevede più servizi a bordo
5	Ricerca di tutti i voli sotto la guida di un pilota specifico
6	Contare il numero di aerei per ogni produttore
7	Visualizzare i passeggeri che hanno prenotato un posto “Business”
8	Trovare la città con il maggior numero di aeroporti
9	Visualizzare i voli in partenza da un determinato aeroporto
10	Visualizzare i voli in arrivo in un determinato aeroporto

Progettazione concettuale

Ho preso la decisione di adottare la strategia "**top-down**" per il mio progetto.

L'approccio "top-down" si basa sull'**analisi** sistematica e **dettagliata** dell'**obiettivo generale** del progetto, per poi suddividerlo in **componenti più piccole** e specifiche. In altre parole, si parte dalla **visione d'insieme** per poi scendere gradualmente nei **dettagli**. Questo permette di avere una **visione chiara** e strutturata del progetto fin dall'**inizio**, focalizzandosi sulle parti più rilevanti e rilevando eventuali problemi o sfide che potrebbero emergere durante l'implementazione.

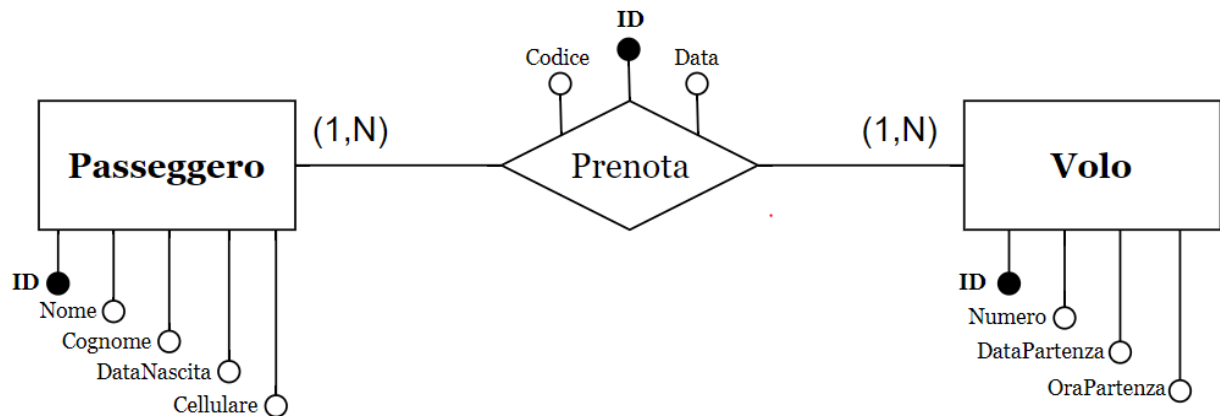
Schema Scheletro



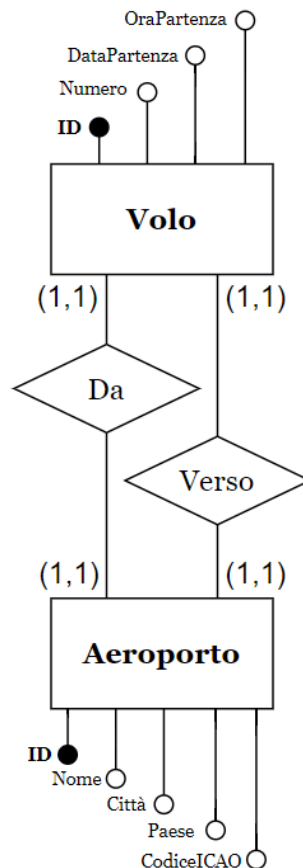
Raffinamenti

Seguono una serie di raffinamenti dello schema scheletro sino ad arrivare ad uno schema finale e completo.

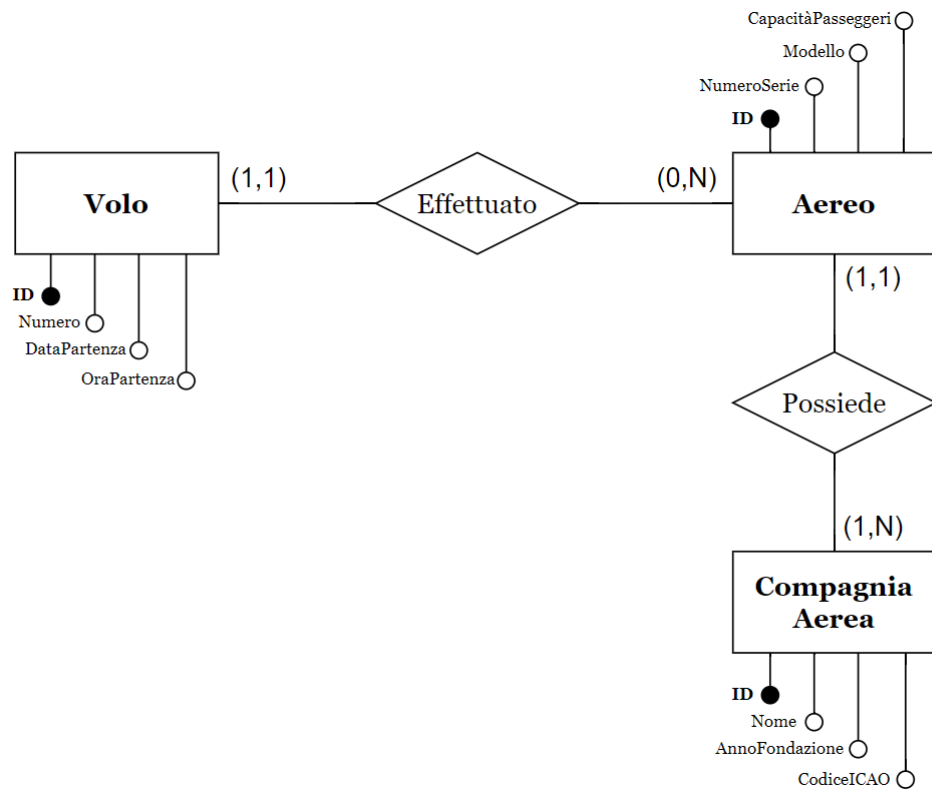
Raffinamento n.1: Passeggero, Prenotazione, Volo



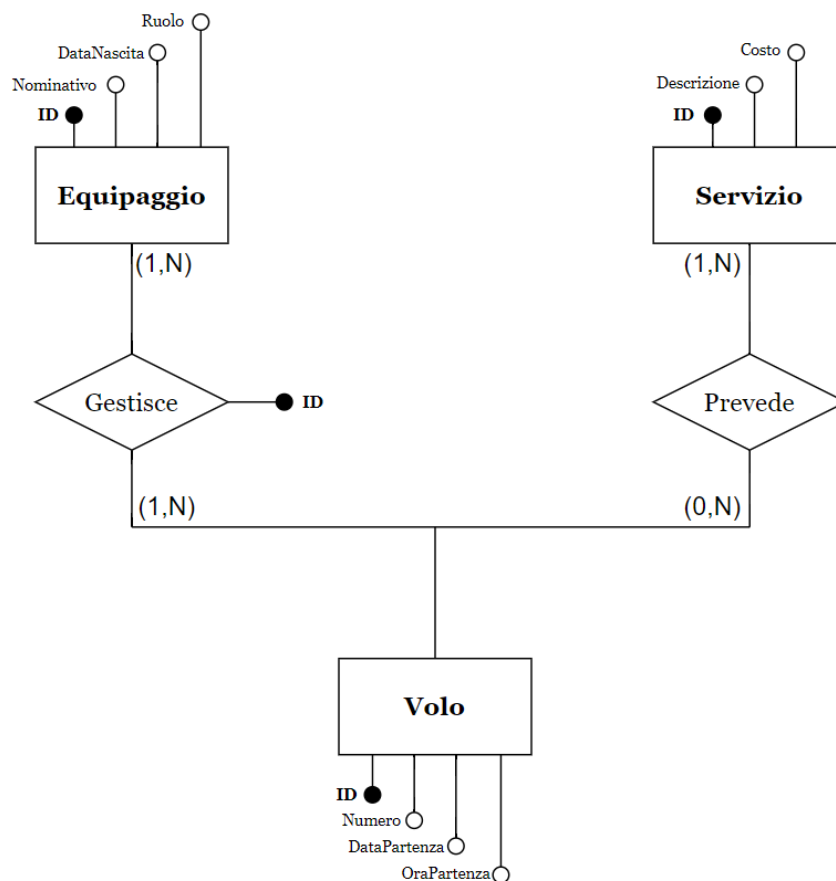
Raffinamento n.2: Volo, Aeroporto



Raffinamento n.3: Volo, Aereo, Compagnia Aerea



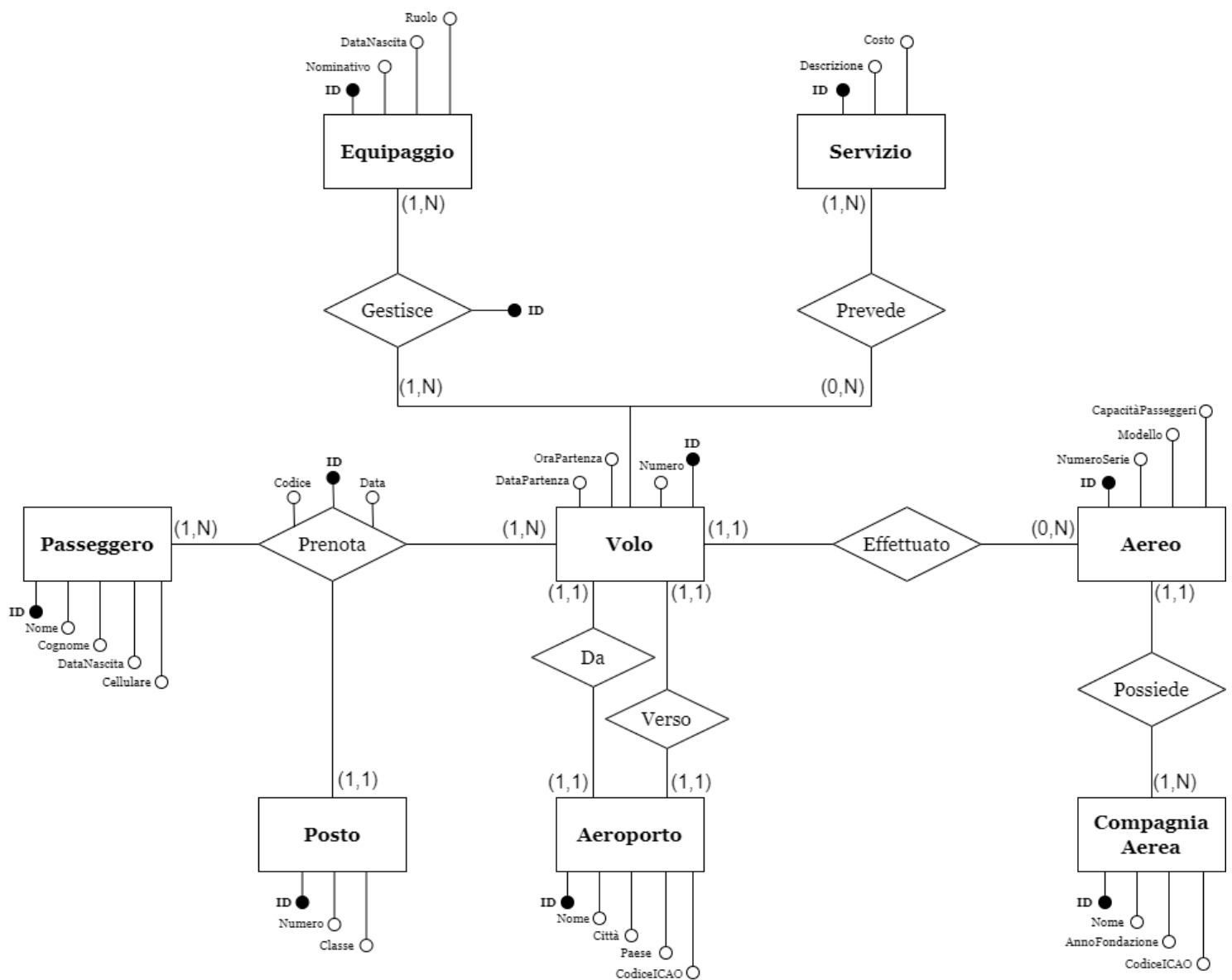
Raffinamento n.4: Volo, Equipaggio, Servizio



Schema Finale

Dopo aver eseguito i raffinamenti sulle cardinalità tra le tabelle presenti nel database, il risultato di tali aggiustamenti è presentato di seguito.

Si sottolinea che tali raffinamenti hanno lo scopo di fornire una **visione più precisa delle relazioni** tra le entità coinvolte nel modello di dati.



Vincoli non esprimibili dal diagramma E-R

- Attributo **classe** nell'entità **Posto**: Indica la tipologia del posto selezionato in fase di prenotazione di un volo, questo attributo può assumere solo uno tra i seguenti valori:
 - Economy
 - Business
 - First
- Attributo **ruolo** nell'entità **Equipaggio**: Indica la tipologia del personale di bordo che opera presso un determinato volo, questo attributo può assumere solo uno tra i seguenti valori:
 - Pilota
 - Copilota
 - Assistente di volo

Inoltre, secondo le attuali normative in vigore, è **obbligatorio** che ogni volo sia equipaggiato con un **pilota** e un **copilota**.

Dizionario dei dati

Entità	Descrizione	Attributi	Identificatore
Aeroporto	Località di decollo ed atterraggio dei voli	Nome, Città, Paese, Codice ICAO	ID
Compagnia Aerea	Ditta che offre i servizi di trasporto aereo	Nome, Anno Fondazione, Codice ICAO	ID
Passeggero	Viaggia a bordo di un aereo	Nome, Cognome, Data Nascita, Cellulare	ID
Volo	Operazione di trasporto aereo	Numero, Data Partenza, Ora Partenza	ID
Aereo	Mezzo di trasporto	Numero Serie, Modello, Capacità passeggeri	ID
Posto	Area di un aereo designata alla seduta dei passeggeri	Numero, Classe	ID
Equipaggio	Gruppo di persone che opera e gestisce l'aereo	Nome, Cognome, Data Nascita, Ruolo	ID
Servizio	Servizi destinati al comfort dei passeggeri	Descrizione, Costo	ID

Dizionario delle relazioni

Associazione	Entità Partecipanti	Descrizione	Attributi
Prenota	Passeggero, Posto, Volo	Acquisto di un viaggio	ID , Codice, Data
Da	Volo, Aeroporto	Aeroporto in fase di decollo	-
Verso	Volo, Aeroporto	Aeroporto in fase di atterraggio	-
Effettuato	Volo, Aereo	Indica con quale aereo è stato effettuato un volo	-
Possiede	Aereo, Compagnia Aerea	Indica a quale compagnia aerea appartiene un aereo	-
Gestisce	Equipaggio, Volo	Assegna un membro dell'equipaggio ad un volo	ID
Prevede	Servizio, Volo	Indica quali servizi sono disponibili in un volo	-

Tabella dei Volumi

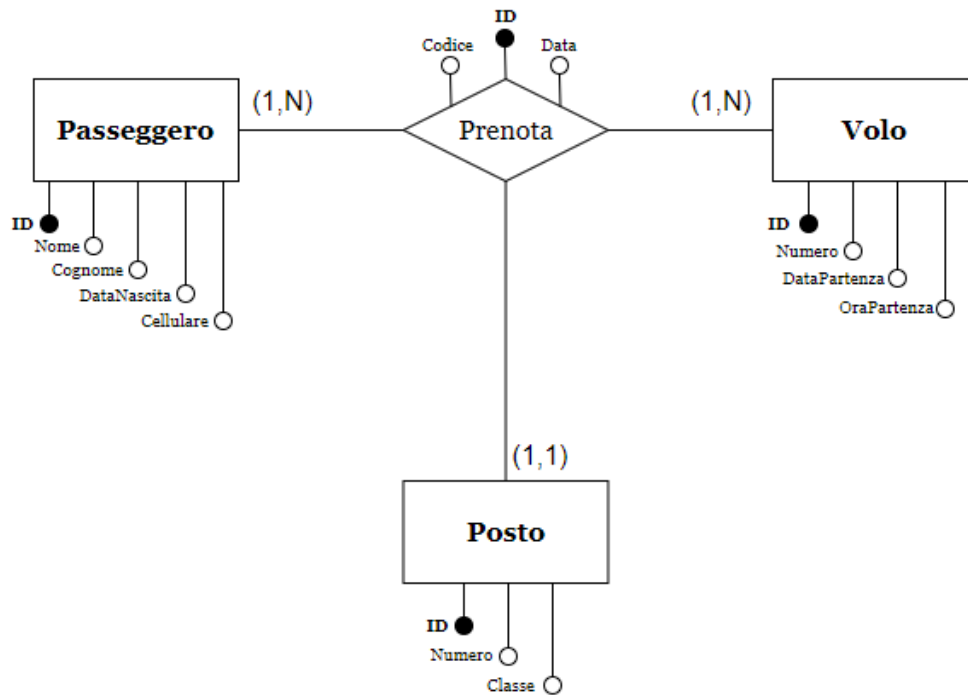
Concetto	Tipo	Volume
Aeroporto	E	50
Compagnia Aerea	E	10
Passeggero	E	500.000
Volo	E	100.000
Aereo	E	3.000
Posto	E	300.000
Equipaggio	E	2.000
Servizio	E	10
Prenota	R	40.000
Da	R	1
Verso	R	1
Effettuato	R	1
Possiede	R	1
Gestisce	R	2.000
Prevede	R	1

Tabella delle Frequenze

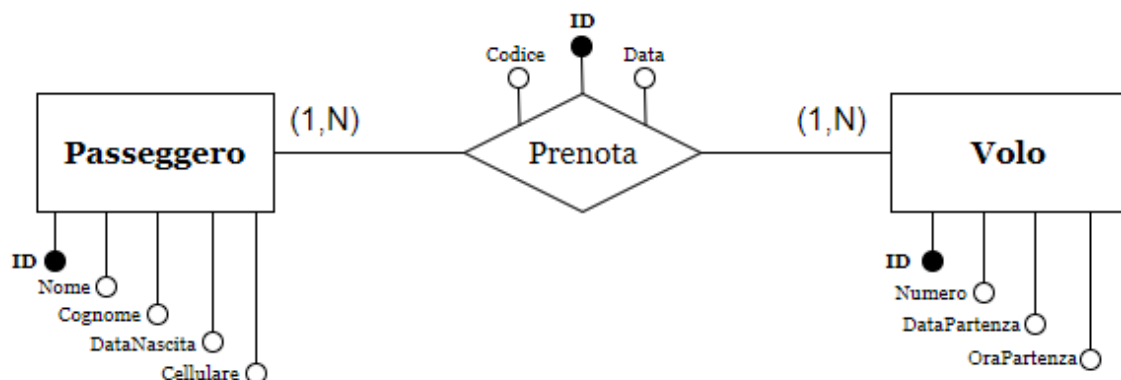
Indice	Tipo	Frequenza
1	I	10/Giorno
2	I	4/Mese
3	B	1/Anno
4	I	1/Mese
5	I	4/Settimana
6	B	4/Anno
7	I	1/Anno
8	I	2/Anno
9	I	4/Giorno
10	I	4/Giorno

Schemi delle Operazioni

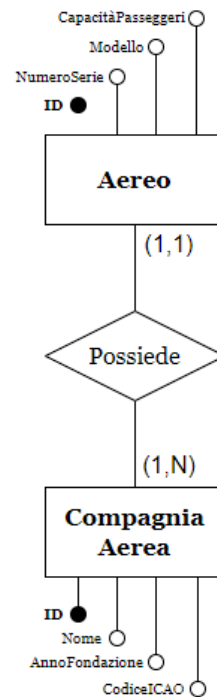
1. Inserire una nuova **prenotazione**



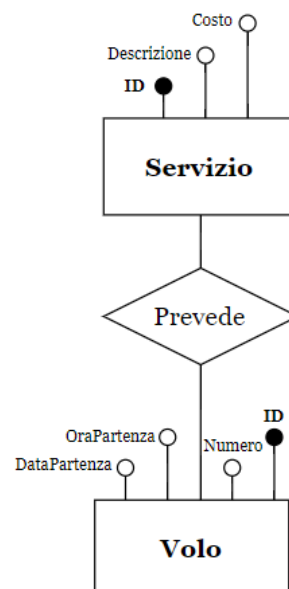
2. Visualizzare i **voli prenotati** da un **passeggero**



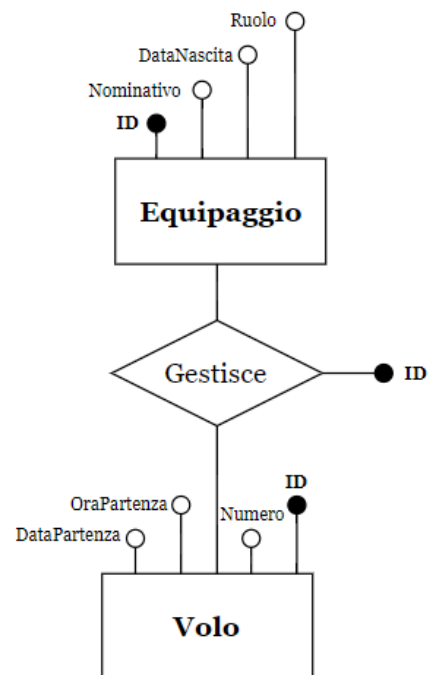
3. Vedere il numero di **aerei** per **compagnia aerea**



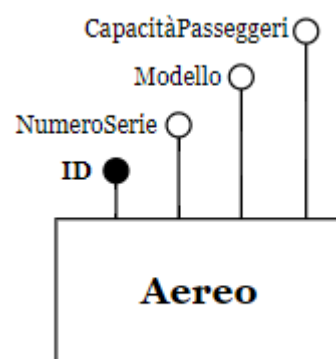
4. Trovare il **volo** che prevede più **servizi** a bordo



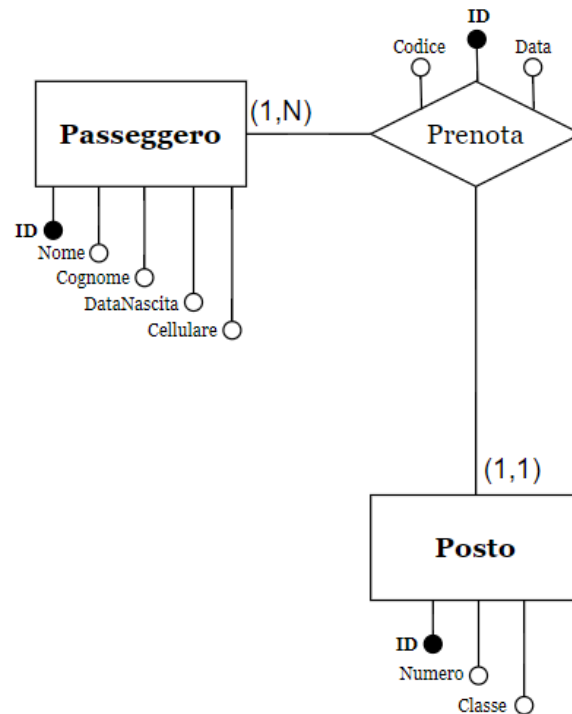
5. Ricerca di tutti i **voli** sotto la guida di un **pilota** specifico



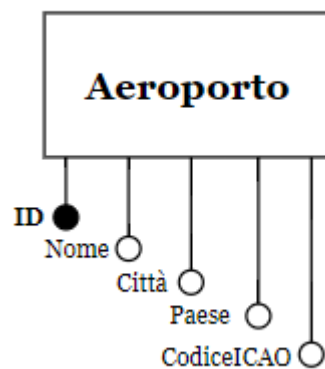
6. Contare il numero di **aerei** per ogni **produttore**



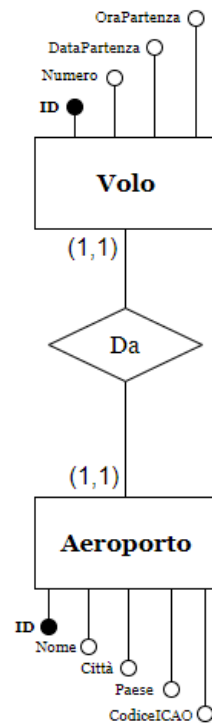
7. Visualizzare i **passaggeri** che hanno prenotato un posto **“Business”**



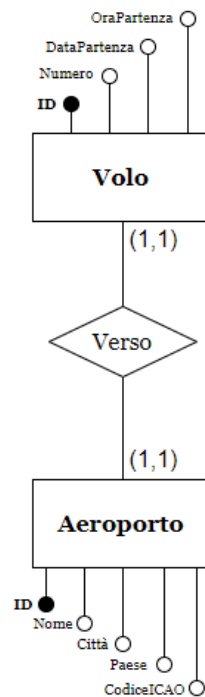
8. Trovare la città con il maggior numero di **aeroporti**



9. Visualizzare i **voli** in **partenza** da un determinato **aeroporto**



10. Visualizzare i **voli** in **arrivo** in un determinato **aeroporto**



Traduzione verso il modello Relazionale

Per costruire un buon database, è essenziale comprendere le **relazioni** tra le diverse entità e tradurle in modo accurato.

Questo ci permette di creare uno schema logico **coerente** che **rifletta la realtà**. Solo così riusciremo a gestire i dati in modo efficace e ottenere informazioni utili dal database.

Tradurre correttamente le **associazioni** tra le varie entità è dunque un passaggio fondamentale per avere un **database funzionale** e di **valore**.

Traduzione delle associazioni

- L'associazione **Prenota** viene concretizzata nell'entità **Prenotazione**.
- L'associazione **Da** viene acquisita nell'entità **Volo** sotto l'attributo "**Aeroporto_Partenza**".
- L'associazione **Verso** viene acquisita nell'entità **Volo** sotto l'attributo "**Aeroporto_Arrivo**".
- L'associazione **Effettuato** viene acquisita nell'entità **Volo** sotto l'attributo "**ID_Aereo**".
- L'associazione **Possiede** viene acquisita nell'entità **Aereo** sotto l'attributo "**Compagnia**".
- L'associazione **Gestisce** viene concretizzata nell'entità **Assegnazione Equipaggio**.
- L'associazione **Prevede** viene acquisita nell'entità **Servizio** sotto l'attributo "**ID_Volo**".

Schema Logico

Lo **schema logico** è una rappresentazione dettagliata e strutturata delle tabelle, delle relazioni e dei vincoli che definiscono la struttura del database. Risulta essere dunque **essenziale** per la creazione di una base di dati ben strutturata, **coerente** e **ottimizzata**.

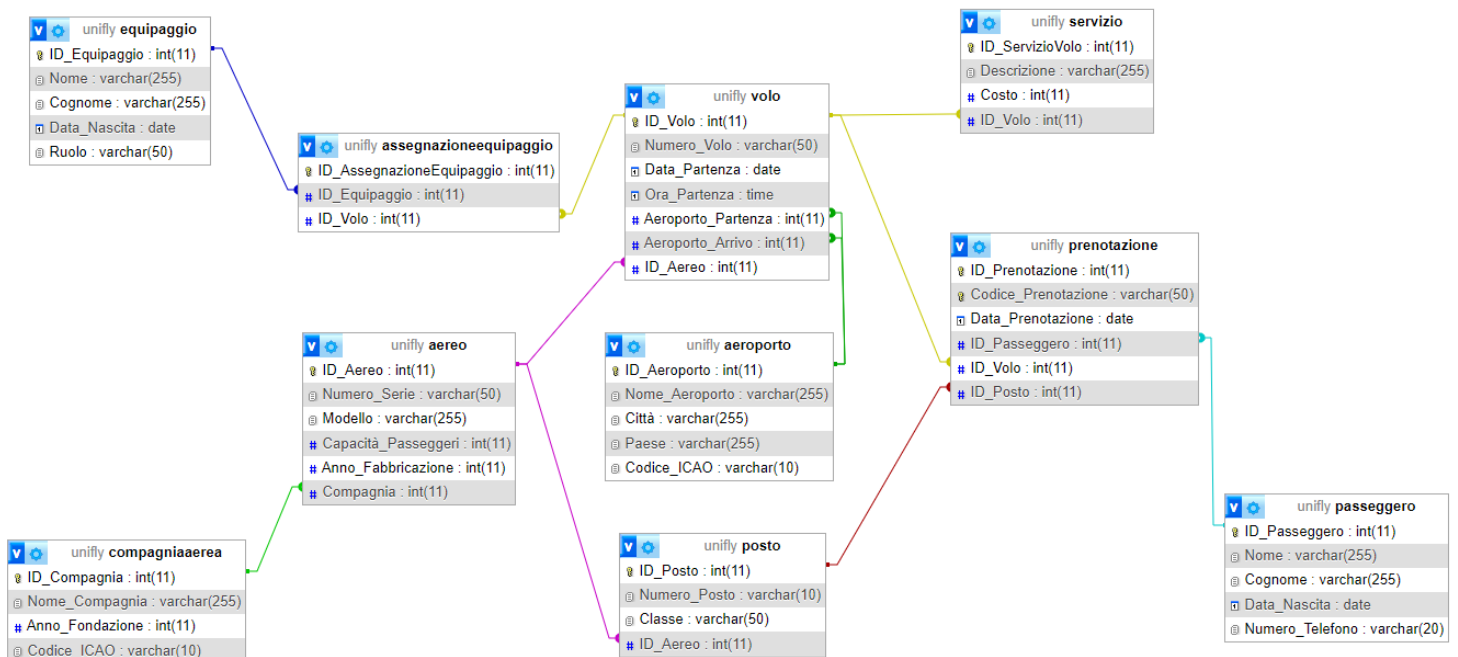
- **Aeroporto** (ID_Aeroporto, Nome_Aeroporto, Città, Paese, Codice_ICAO)
- **CompagniaAerea** (ID_Compagnia, Nome_Compagnia, Anno_Fondazione, Codice_ICAO)
- **Passeggero** (ID_Passeggero, Nome, Cognome, Data_Nascita, Numero_Telefono)
- **Aereo** (ID_Aereo, Numero_Serie, Modello, Capacità_Passeggeri, Anno_Fabbricazione, [Compagnia](#))
- **Volo** (ID_Volo, Numero_Volo, Data_Partenza, Ora_Partenza, [Aeroporto_Partenza](#), [Aeroporto_Arrivo](#), [ID_Aereo](#))
- **Posto** (ID_Posto, Numero_Posto, Classe, [ID_Aereo](#))
- **Equipaggio** (ID_Equipaggio, Nome, Cognome, Data_Nascita, Ruolo)
- **AssegnazioneEquipaggio** (ID_AssignazioneEquipaggio , [ID_Equipaggio](#), [ID_Volo](#))
- **Servizio** (ID_ServizioVolo, Descrizione, Costo, [ID_Volo](#))
- **Prenotazione** (ID_Prenotazione, Codice_Prenotazione, Data_Prenotazione, [ID_Passeggero](#), [ID_Volo](#), [ID_Posto](#))

Legenda

- NomeChiave: Chiave Primaria (PK)
- [NomeChiave](#): Chiave Esterna (FK)

Schema Fisico

Lo **schema fisico** di un database rappresenta la struttura di memorizzazione dei dati sul **sistema di archiviazione** sottostante. Contrariamente allo schema logico, che descrive la struttura dati a un livello concettuale e indipendente dal sistema di gestione di database, lo schema fisico **definisce** come **i dati** vengono effettivamente organizzati, memorizzati e gestiti nel **DBMS**.



Creazione di un trigger SQL

UniFly, per garantire la massima tutela ai propri clienti, ha deliberato di non adottare l'**overbooking** come pratica commerciale.

Pertanto, ogni tentativo di prenotazione per un volo che risulta essere completamente prenotato sarà impedito dal **trigger** denominato **NoOverbooking**. In tali circostanze, l'utente riceverà un messaggio di errore e l'inserimento non sarà completato con successo.

```
CREATE TRIGGER `NO_Overbooking`
BEFORE INSERT ON `prenotazione` FOR EACH ROW BEGIN
    DECLARE total_posti_prenotati INT;
    DECLARE capacita_aereo INT;

    -- Calcola il numero di posti già prenotati per il volo della nuova
    prenotazione
    SELECT COUNT(ID_Prenotazione) INTO total_posti_prenotati
    FROM Prenotazione
    WHERE ID_Volo = NEW.ID_Volo;

    -- Ottieni la capacità massima dell'aereo del volo della nuova
    prenotazione
    SELECT Capacità_Passeggeri INTO capacita_aereo
    FROM Aereo
    WHERE ID_Aereo = (
        SELECT ID_Aereo
        FROM Volo
        WHERE ID_Volo = NEW.ID_Volo
    );

    -- Se il numero di posti prenotati supera la capacità dell'aereo,
    genera un errore
    IF total_posti_prenotati >= capacita_aereo THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Numero massimo di posti prenotati superato
    per questo volo.';
    END IF;
END
```


Nel caso in cui il trigger "**NoOverbooking**" venga attivato, verrà visualizzato un **messaggio di errore** personalizzato per notificare all'utente che l'inserimento della prenotazione non è andato a buon fine a causa dell'esaurimento dei posti disponibili sul volo desiderato.

Il messaggio di errore è il **seguito**:



Creazione di una vista in SQL

In SQL, una **vista** è una **tabella virtuale** che rappresenta una visualizzazione dei dati estratti da una query definita dall'utente da una o più tabelle.

Nell'esempio sottostante, si vuole visualizzare il numero di **prenotazioni** effettuate per ogni **volo** presente nel DBMS.

```
CREATE VIEW `prenotazionipervolo` AS
SELECT ID_Volo, count(ID_Prenotazione) AS `NumeroPrenotazioni`
FROM prenotazione
GROUP BY ID_Volo;
```

Query in SQL

Inserire una nuova **prenotazione**:

```
INSERT INTO Prenotazione (Codice_Prenotazione, Data_Prenotazione,  
ID_Passeggero, ID_Volo, ID_Posto)  
VALUES (... , ... , [1,30], [1,30], [1,50]);
```

Per effettuare le modifiche ai valori presenti tra parentesi, è importante rispettare i **range** indicati. Si prega di prestare attenzione a non superare i limiti imposti e di mantenere la corretta coerenza dei valori.

Visualizzare i **voli prenotati** da un **passeggero**:

Visualizza il numero del volo, la data e l'orario di partenza e gli aeroporti relativi alle prenotazioni effettuate da un passeggero.

```
SELECT V.ID_Volo, V.Numero_Volo, V.Data_Partenza, V.Ora_Partenza,  
V.Aeroporto_Partenza, V.Aeroporto_Arrivo  
FROM Volo V  
JOIN Prenotazione P ON V.ID_Volo = P.ID_Volo  
WHERE P.ID_Passeggero = <?>;
```

<?>: Inserisci **IDPasseggero** [1,30]

Contare il **numero** di **aerei** per ogni compagnia **aerea**:

Visualizza il nome della compagnia aerea ed il numero di aerei in suo possesso.

```
SELECT CompagniaAerea.ID_Compagnia, CompagniaAerea.Nome_Compagnia,  
COUNT(Aereo.ID_Aereo) AS Numero_Aerei  
FROM CompagniaAerea  
JOIN Aereo ON CompagniaAerea.ID_Compagnia = Aereo.Compagnia  
GROUP BY CompagniaAerea.ID_Compagnia, CompagniaAerea.Nome_Compagnia;
```

Trovare il **volo** che prevede più **servizi** a bordo:

Visualizza l'ID del volo che prevede il numero maggiore di servizi offerti.

```
SELECT ID_Volo, COUNT(ID_ServizioVolo) AS Numero_Servizi
FROM Servizio
GROUP BY ID_Volo
HAVING COUNT(ID_ServizioVolo) = (SELECT MAX(CountServizi)
                                FROM (SELECT COUNT(ID_ServizioVolo)
                                      AS CountServizi
                                      FROM Servizio
                                      GROUP BY ID_Volo) AS CountTable);
```

Ricerca di tutti i **voli** sotto la guida di un **pilota** specifico:

Visualizza l'ID del volo, il numero del volo, la data e l'ora di partenza e gli aeroporti di riferimento.

```
SELECT V.ID_Volo, V.Numero_Volo, V.Data_Partenza, V.Ora_Partenza,
V.Aeroporto_Partenza, V.Aeroporto_Arrivo
FROM Volo V
JOIN AssegnazioneEquipaggio AE ON V.ID_Volo = AE.ID_Volo
JOIN Equipaggio E ON AE.ID_Equipaggio = E.ID_Equipaggio
WHERE E.ID_Equipaggio = <?>;
```

<?>: Inserisci **IDEquipaggio** [1,12]

Contare il **numero** di **aerei** per ogni **produttore**:

Visualizza il nome del produttore ed il numero di aerei da esso prodotti.

```
SELECT Numero_Serie as Produttore, COUNT(ID_Aereo) AS Numero_Aerei
FROM Aereo
GROUP BY Produttore;
```

Visualizzare i **passengeri** che hanno prenotato un **posto** “*Business*”:

Visualizza il nome ed il cognome dei passeggeri che hanno prenotato un posto di tipo Business.

```
SELECT P.ID_Passeggero, P.Nome, P.Cognome
FROM Prenotazione PR
JOIN Posto PO ON PR.ID_Posto = PO.ID_Posto
JOIN Passeggero P ON PR.ID_Passeggero = P.ID_Passeggero
WHERE PO.Classe = 'Business';
```

Trovare la **città** con il maggior numero di **aeroporti**:

Visualizza il nome della città con il maggior numero di aeroporti presenti al suo interno.

```
SELECT Città, COUNT(ID_Aeroporto) AS Numero_Aeroporti
FROM Aeroporto
GROUP BY Città
ORDER BY Numero_Aeroporti DESC
LIMIT 1;
```

Visualizzare i voli in **partenza** da un determinato **aeroporto**:

Visualizza il numero del volo, la data e l'orario di partenza e gli aeroporti di riferimento dei voli che partono da un determinato aeroporto.

```
SELECT V.ID_Volo, V.Numero_Volo, V.Data_Partenza, V.Ora_Partenza,
V.Aeroporto_Partenza, V.Aeroporto_Arrivo
FROM Volo V
JOIN Aeroporto AP ON V.Aeroporto_Partenza = AP.ID_Aeroporto
WHERE AP.ID_Aeroporto = <?>;
```

<?>: Inserisci IDAeroporto [1,10]